Full credit is given to the above companies including the OS that this PDF file was generated!

Red Hat Enterprise Linux Release 9.2 Manual Pages on 'RSA_check_key_ex.3ossl' command

$ man RSA_check_key_ex.3ossl

RSA_CHECK_KEY(3ossl) OpenSSL RSA_CHECK_KEY(3ossl)

NAME

RSA_check_key_ex, RSA_check_key - validate private RSA keys

SYNOPSIS

#include <openssl/rsa.h>

The following functions have been deprecated since OpenSSL 3.0, and can be hidden entirely by defining OPENSSL_API_COMPAT with a suitable version value, see openssl_user_macros(7):

int RSA_check_key_ex(const RSA *rsa, BN_GENCB *cb);

int RSA_check_key(const RSA *rsa);

DESCRIPTION

Both of the functions described on this page are deprecated.

Applications should instead use EVP_PKEY_public_check(3), EVP_PKEY_private_check(3) and EVP_PKEY_pairwise_check(3).

RSA_check_key_ex() function validates RSA keys. It checks that p and q are in fact prime, and that n = p*q.

It does not work on RSA public keys that have only the modulus and public exponent elements populated. It also checks that d*e = 1 mod (p-1*q-1), and that dmp1, dmq1 and iqmp are set correctly or are NULL.

It performs integrity checks on all the RSA key material, so the RSA key structure must contain all the private key data too. Therefore, it cannot be used with any arbitrary RSA key object, even if it is otherwise fit for regular RSA operation.

The cb parameter is a callback that will be invoked in the same manner as BN_is_prime_ex(3).

RSA_check_key() is equivalent to RSA_check_key_ex() with a NULL cb.

RETURN VALUES

RSA_check_key_ex() and RSA_check_key() return 1 if rsa is a valid RSA key, and 0 otherwise. They return -1 if an error occurs while checking the key.

If the key is invalid or an error occurred, the reason code can be obtained using ERR_get_error(3).

NOTES

Unlike most other RSA functions, this function does not work transparently with any underlying ENGINE implementation because it uses the key data in the RSA structure directly. An ENGINE implementation can override the way key data is stored and handled, and can even provide support for HSM keys - in which case the RSA structure may contain no key data at all! If the ENGINE in question is only being used for acceleration or analysis purposes, then in all likelihood the RSA key data is complete and untouched, but this can't be assumed in the general case.

BUGS

A method of verifying the RSA key using opaque RSA API functions might need to be considered. Right now RSA_check_key() simply uses the RSA structure elements directly, bypassing the RSA_METHOD table altogether (and completely violating encapsulation and object-orientation in the process). The best fix will probably be to introduce a "check_key()" handler to the RSA_METHOD function table so that alternative implementations can also provide their own verifiers.

SEE ALSO

BN_is_prime_ex(3), ERR_get_error(3)

HISTORY

All of these functions were deprecated in OpenSSL 3.0.

RSA_check_key_ex() appeared after OpenSSL 1.0.2.

COPYRIGHT